CMPSC 465 — Data Structures & Algorithms

Spring 2024 — Mehrdad Mahdavi and David Koslicki

Worksheet 1

Wednesday, January 17, 2024

1. **Compare Growth Rates.** Order the following functions by asymptotic growth:

   (i) $f_1(n) = 3^n$

   (ii) $f_2(n) = n^{\frac{1}{3}}$

   (iii) $f_3(n) = 12$

   (iv) $f_4(n) = 2^{\log_2 n}$

   (v) $f_5(n) = \sqrt{n}$

   (vi) $f_6(n) = 2^n$

   (vii) $f_7(n) = \log_2 n$

   (viii) $f_8(n) = 2^{\sqrt{n}}$

   (ix) $f_9(n) = n^3$

   Solution $f_3, f_7, f_2, f_5, f_4, f_9, f_8, f_6, f_1$

   - $f_3 = O(f_7)$: By the definition of Big-O, let $c = 12$ and $n_0 = 2$. Then $12 \leq c\log_2 n$ for all $n \geq n_0$.

   - $f_7 = O(f_2)$: You could use the fact that any polynomial dominates any logarithm. Without this, you could also prove it by using basic facts about the log function and the definition of Big O as follows. From the definition of O it suffices to show that $\log_2 n \leq 3n^{1/3}$, which is equivalent to $\log_2 n^{1/3} \leq n^{1/3}$, and this follows from the fact that $\log_2 x \leq x$ for all $x > 0$; the latter is true because $x - \log_2 x$ is convex with a derivative of $1 - \frac{1}{x\ln 2}$ which is 0 at $x = \frac{1}{\ln 2}$ which means $\min_x x - \log_2 x = \frac{1}{\ln 2} + \log_2 \ln 2 = 0.9139 \geq 0$

   - $f_2 = O(f_5)$: $\sqrt{n} = n^{\frac{1}{2}}$. Then, it suffices to observe that $n^{1/3} \leq n^{1/2}$ since $\frac{1}{2} > \frac{1}{3}$.

   - $f_5 = O(f_4)$: $2^{\log_2 n} = n = n^1$. As $1 > \frac{1}{2}$, $f_4 \geq f_5$.

   - $f_4 = O(f_9)$: $3 > 1$, so $f_9 \geq f_4$.

   - $f_9 = O(f_8)$: You could use the fact that any exponential dominates any polynomial. Alternatively, you could prove this from first principles, similar to the way f_7 was shown to be $O(f_2)$ (check this!).

- $f_8 = O(f_6)$: By definition. Take $c = 1$ and $n_0 = 1$. $2^{\sqrt{n}} \leq 2^n$ for all $n \geq 1$.

- $f_6 = O(f_1)$: By definition. Take $c = 1$ and $n_0 = 1$. $2^n \leq 3^n$ for all $n \geq 1$.

  Note that unlike logarithm bases, which do not affect the growth rate of the logarithm, larger exponent bases grow faster than smaller exponent bases. This can be shown with the following limit:

  $$\lim_{n\to\infty} \frac{3^n}{2^n} = \lim_{n\to\infty} (\tfrac{3}{2})^n = \infty$$

  An aside about the equivalent growth rates of logarithms of different bases - this can be explained by following the change of base formula, which shows that any two logarithms of different bases are separated by a constant multiple.

2. Prove Order of Growth. Prove the following:

(i) $\log(n!) = \Theta(n \log n)$

(ii) $\sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$

Solution

(i) Observe that

$$n! = 1 * 2 * 3 \cdots * n \leq n * n * n \cdots * n = n^n$$

and

$$n! = 1 * 2 * 3 \cdots * n \geq n * (n-1) * (n-2) \cdots * (n - \lfloor \tfrac{n}{2} \rfloor) \geq \left(\frac{n}{2}\right)^{\lfloor \frac{n}{2} \rfloor} \geq \left(\frac{n}{2}\right)^{\frac{n}{2}-1}$$

Hence $\left(\frac{n}{2}\right)^{\frac{n}{2}-1} \leq n! \leq n^n$. Then by taking the logarithm:

$$\left(\frac{n}{2} - 1\right) \log\left(\frac{n}{2}\right) \leq \log(n!) \leq n \log n.$$

Finally note that

$$\left(\frac{n}{2} - 1\right) \log\left(\frac{n}{2}\right) \geq \frac{1}{2}\left(\frac{n}{2} - 1\right) \log(n) \geq \left(\frac{n}{8}\right) \log(n).$$

Where both of these inequalities hold for all $n \geq 4$.

$$\frac{1}{8} n \log(n) \leq \log(n!) \leq n \log n.$$

for all $n \geq 4$.

(ii) The main idea for both upper and lower bound is that for any k there exist a unique value of i such that $\frac{1}{2^{i+1}} \leq \frac{1}{k} \leq \frac{1}{2^i}$. Note that for several k's the value of i is the same, so we can replace several $1/k$'s by the same inverse power of 2 to obtain a more manageable sum. More precisely, for the upper bound we replace each $1/k$ with $1/2^i$, where 2^i is the largest power of 2 less than or equal to k. Let $\ell = \lceil \log_2 n \rceil$.

$$\begin{aligned}
\sum_{k=1}^{n} \frac{1}{k} &= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \\
&\leq 1 + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \cdots \\
&\leq 1 + 2 \cdot \tfrac{1}{2} + 4 \cdot \tfrac{1}{4} + \cdots + 2^{\ell} \cdot \frac{1}{2^{\ell}}. \\
&= 1 + 1 + 1 + \cdots + 1 = \ell + 1 = O(\log n)
\end{aligned}$$

There are $\ell + 1$ terms in the final sum because there is a single 1 for each power of 2 from 0 to ℓ. This is always an upper bound, regardless of whether or not n is a power of 2, due to including all 2^{ℓ} of the $\frac{1}{2^{\ell}}$ terms. At most one of these terms is present in the original sum, since $n \leq 2^{\ell}$. We can do something similar for the lower bound, except we replace $1/k$ with $1/2^i$, where 2^i

is the smallest power of 2 greater than k. We can guarantee this sum is a lower bound by only including terms up to $\frac{1}{2^{\ell-1}}$.

$$\sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \geq \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \cdots \geq \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \cdots = \frac{\ell - 1}{2} = \Omega(\log n)$$

Note. This problem can also be solved by integrating: $\sum_{k=1}^{n} \frac{1}{k} = \Theta(\int_1^n \frac{1}{x} dx) = \Theta(\log n)$.

3. Analyze Running Time. For each pseudo-code below, give the asymptotic running time in Θ notation.

(i)

```
for i := 1 to n do
    j := i;
    while j < n do
        j := j + 5;
    end
end
```

Solution Inside the inner loop, a constant number of basic steps are performed. For any fixed i, the inner loop performs $\lceil \frac{n-i}{5} \rceil$ iterations. A lower bound for the total running time is:

$$\sum_{i=1}^{n} \lceil \tfrac{n-i}{5} \rceil \geq \sum_{i=1}^{n} \frac{n-i}{5} = \frac{1}{5}\sum_{i=1}^{n}(n-i) = \frac{1}{5}\sum_{i=0}^{n-1} i = \frac{1}{5}(\tfrac{(n-1)(n)}{2}) = \Omega(n^2).$$

And an upper bound (using the fact that $\lceil x \rceil \leq x+1$) is:

$$\sum_{i=1}^{n} \lceil \tfrac{n-i}{5} \rceil \leq \sum_{i=1}^{n} (\frac{n-i}{5}+1) = \frac{1}{5}\sum_{i=1}^{n}(n-i) + \sum_{i=1}^{n} 1 = \frac{1}{5}\sum_{i=0}^{n-1}(i) + n = \frac{1}{5}(\tfrac{(n-1)(n)}{2}) + n = O(n^2).$$

Therefore, the asymptotic running time is $\Theta(n^2)$.

A faster way to arrive at the solution is observing that $\lceil \frac{n-i}{5} \rceil = \Theta(n-i)$. This means that for every fixed i, there is some coefficients a_i and b_i such that $a_i(n-i) \leq \lceil \frac{n-i}{5} \rceil \leq b_i(n-i)$ for all n greater than some n_0. If we instead use the minimum a_{min} and the maximum b_{max} among these coefficients, we can ensure that for every term, $a_{min}(n-i) \leq \lceil \frac{n-i}{5} \rceil \leq b_{max}(n-i)$. Applying this to the entire sum using the distributive property, we get:

$$a_{min}\sum_{i=1}^{n}(n-i) \leq \sum_{i=1}^{n} \lceil \tfrac{n-i}{5} \rceil \leq b_{max}\sum_{i=1}^{n}(n-i)$$

$$a_{min}\sum_{i=0}^{n-1} i \leq \sum_{i=1}^{n} \lceil \tfrac{n-i}{5} \rceil \leq b_{max}\sum_{i=0}^{n-1} i$$

$$a_{min}(\tfrac{(n-1)(n)}{2}) \leq \sum_{i=1}^{n} \lceil \tfrac{n-i}{5} \rceil \leq b_{max}(\tfrac{(n-1)(n)}{2})$$

$$\sum_{i=1}^{n} \lceil \tfrac{n-i}{5} \rceil = \Theta(n^2)$$

Applying the idea of distributing the minimum and maximum coefficients to the general case, we can see that:

$$a_{min}\sum_{i=1}^{n} f(i) \leq \sum_{i=1}^{n} \Theta(f(i)) \leq b_{max}\sum_{i=1}^{n} f(i)$$

$$\sum_{i=1}^{n} \Theta(f(i)) = \Theta(\sum_{i=1}^{n} f(i))$$

(ii)
```
for i := 1 to n do
    for j := 4i to n do
        s := s + 2;
    end
end
```

Solution For any fixed $i \le n/4$, the inner loop iterates $n-4i+1$ times, and zero times when $i > n/4$. Then, the running time is

$$\frac{3n}{4}+\sum_{i=1}^{n/4}(n-4i+1)=\frac{3n}{4}+\sum_{i=1}^{n/4}n-4\sum_{i=1}^{n/4}i+n/4=\frac{3n}{4}+\frac{n^2}{4}-4\cdot\frac{\frac{n}{4}(\frac{n}{4}+1)}{2}+n/4=\Theta(n^2).$$

Note that a constant amount of work is still done by the outer loop even when the inner loop does not iterate (and the inner loop when considering that it must check that $i > n/4$).

(iii)
```
for i := 1 to n do
    j := 2;
    while j < i do
        j := j^4;
    end
end
```

Solution For each fixed $i > 2$, the inner loop iterates at most $\log_4 \log_2 i + 1 = \Theta(\log\log i)$ times, since the value of j in the k-th iteration of the inner loop is $2^{4^{k-1}}$, and it runs while $2^{4^{k-1}} < i$. Hence, the running time is $O(1)+\sum_{i=3}^{n}\Theta(\log\log i)=\Theta(\sum_{i=3}^{n}\log\log i)$.

$$\sum_{i=3}^{n}\log\log i \le \sum_{i=3}^{n}\log\log n = n\log\log n = O(n\log\log n)$$

$$\sum_{i=3}^{n}\log\log i \ge \sum_{i=\lceil n/2\rceil}^{n}\log\log i \ge \sum_{i=\lceil n/2\rceil}^{n}\log\log\frac{n}{2} \ge \lfloor\tfrac{n}{2}\rfloor\log\log\tfrac{n}{2} = \Omega(n\log\log n),$$

where the first inequality holds for $n \ge 6$ say. Note that this style of proof can be applied to the summation of any asymptotically increasing function f. The lower bound additionally requires that $f(\frac{n}{2}) = \Theta(f(n))$, which is the case for logarithms and polynomials, but not exponentials. Bounding the summation of an exponential is included in homework 1. For this upper bound, you replace all the terms with $f(n)$:

$$\sum_{i=1}^{n}f(i) \le \sum_{i=1}^{n}f(n) = nf(n) = O(nf(n))$$

For the lower bound, you take only the upper half of the terms, then replace them all with $f(\frac{n}{2})$:

$$\sum_{i=1}^{n}f(i) \ge \sum_{i=\lceil n/2\rceil}^{n}f(i) \ge \sum_{i=\lceil n/2\rceil}^{n}f(\tfrac{n}{2}) \ge \lfloor\tfrac{n}{2}\rfloor f(\tfrac{n}{2}) = \Omega(nf(n))$$

These bounds show that $\sum_{i=1}^{n}f(i)=\Theta(nf(n))$.

Using this and the fact that $\sum_{i=1}^{n} \Theta(f(i)) = \Theta(\sum_{i=1}^{n} f(i))$ established in part (i), we see that for any asymptotically increasing function f that has $f(\frac{n}{2}) = \Theta(f(n))$:

$$\sum_{i=1}^{n} \Theta(f(i)) = \Theta(nf(n))$$

4. Polynomial and Exponential Growth. Prove the following:

$$n^c = O(a^n) \quad \forall c > 0, a > 1$$

Solution First, we show that $n^c = O(a^n)$ can be expressed as $n = O(k^n)$ for some $k > 1$. To do this, we take the c-th root of both sides:

$$n^c = O(a^n) \implies (n^c)^{\frac{1}{c}} = O((a^n)^{\frac{1}{c}}) \implies n = O((a^{\frac{1}{c}})^n) \implies n = O(k^n)$$

Where $k = a^{\frac{1}{c}} > 1$ because $a > 1$ and $c > 0$. This is an equivalent way to write the given equation using only one constant instead of two.

To show that $n = O(k^n)$ for all $k > 1$, consider rewriting k^n using $k = 1 + \varepsilon$:

$$k^n = (1+\varepsilon)^n$$

$(1+\varepsilon)^n$ can be expanded using the binomial formula:

$$(1+\varepsilon)^n = \sum_{i=0}^{n} \binom{n}{i} \varepsilon^i 1^{n-i} = \sum_{i=0}^{n} \binom{n}{i} \varepsilon^i = \binom{n}{0} \cdot \varepsilon^0 + \binom{n}{1} \cdot \varepsilon^1 + \binom{n}{2} \cdot \varepsilon^2 + \cdots + \binom{n}{n} \cdot \varepsilon^n$$

Taking just one term from the sum as a lower bound, for all $n > 1$:

$$k^n \geq \binom{n}{1} \cdot \varepsilon^1 \geq n\varepsilon$$

Letting $c = \frac{1}{\varepsilon}$ and $n_0 = 1$, $n \leq ck^n$ for all $n > n_0$. Therefore, $n = O(k^n)$ and $n^c = O(a^n)$.